10517352

Please replace paragraph [0073] of the published patent application US2005/0261946 with the following paragraph:

[0073] ~~Langrangean~~ Lagrangean relaxation has long been recognised as an effective solution method for constrained optimisation. Many computationally hard problems complicated by a set of difficult constraints can be decomposed into problems with a simpler structure. In our railway scheduling model the track capacity constraints are removed from the constraint set and placed in the objective function by the use of lagrange multipliers. These multipliers can be interpreted as the cost for using the track at a particular time. The higher the price on a track segment at a particular time the less likely it is to be used by trains at that time. This relaxed for of the scheduling problem allows us to reduce the problem to a series of shortest path problems for individual trains on the network. Trains are scheduled through the rail network one at a time through the matrix of prices (Lagrange multipliers) along their least cost path irrespective of other trains in the network. The solution of the relaxed problem with section capacity constraints removed may result in an infeasible schedule. A heuristic method must then be employed to remove the infeasible train movements and produce a feasible schedule.

Please replace paragraph [0084] of the published patent application US2005/0261946 with the following paragraph:

[0084] 3. Form ~~From~~ a contender set of trains consisting of all trains that have as their next move a dispatch from station S_i to S_j and vice-versa.

0095-0096

Please replace paragraph [~~0095]~~ of the published patent application US2005/0261946 with the following paragraph:

6/19/08

[0095] The problem space search dispatcher has been tested on two Australian railway networks. The objective function is that is used to evaluate each schedule is the sum of the lateness of each train. The lateness of each train is given by the function

10517358

$$Z^{i}(a_{id})=\begin{cases}0 & a_{id}\le a_{i}^{*}\\ a_{id}-a_{i}^{*} & a_{id}>a_{i}^{*}\end{cases}$$

where a_{id} is the actual arrival time of train i at is destination while a_i* is the desired arrival time. ~~The problem space search dispatcher was coded in Pascal and run on a Unix workstation.~~

Please replace paragraph [0098] of the published patent application US2005/0261946 with the following paragraph:

[0098] A histogram of the results from both phase I and phase II of the problem space search can be found in figure 3. In both phases 3000 feasible schedules were constructed and the histograms have been plotted using buckets of 1000 seconds. The lowest cost schedule found in phase I had a cost of 152000 seconds while the overall best solution was found in phase II with a cost of 145000 seconds. As can be seen in Figure 3 of the phase II distribution of solution costs has been significantly skewed towards the low cost end when compared with the results from phase I. The best solution found is represented as a train graph in figure 5. Current best practice for this same actual scheduling task on the North Coast line is 205000 seconds. This is shown as the vertical line in figure 3. Our procedure is therefore generating a raft of better schedules, the best being approximately 30% lower than current practice. ~~The program took 10 minutes to run.~~

6/19/05

Please replace paragraph [0100] of the published patent application US2005/0261946 with the following paragraph:

[0100] The results from both phases of the problem space search are presented in figure 4. Once again 3000 feasible schedules were constructed in both phases with the best schedule being found in phase II. The minimum cost schedule found in phase I was 53660 seconds while in phase II the best one found had a cost of 52945 seconds. Note the effect of the weights in skewing the histogram in phase II towards the low cost region. Figure 6 shows the train graph of the best found solution. Current best practice

10517352

for the Sydney to Melbourne scheduling task has a cost of 85000 seconds which is shown as the vertical line in Figure 4. ~~For this larger problem the running time was 29 minutes.~~

Please delete paragraphs [0102] to [~~0108~~ 0109].

6/19/08